UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,722	11/09/2000	Stephan J. Jourdan	2207/9800	2194

23838 7590 09/24/2003

KENYON & KENYON
1500 K STREET, N.W., SUITE 700
WASHINGTON, DC 20005

EXAMINER
O'BRIEN, BARRY J

ART UNIT	PAPER NUMBER
2183	3

DATE MAILED: 09/24/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/708,722	JOURDAN ET AL.	
	Examiner	Art Unit	
	Barry J. O'Brien	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 November 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *09 November 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *2* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-19 have been examined.

Drawings

2. Figure 1 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

4. Claims 1, 6, 9, 11-13, 15-17 and 19 are objected to because of the following informalities:

a. Regarding claim 6, the limitation "instruction cache system" lacks antecedent basis. Please change the limitation "instruction storage system" to read "instruction cache system" to correct the claim language and provide correct antecedent basis. See also claim 15 for a similar correction.

b. Regarding claim 6, please change the phrase "the output" to read "an output" to provide correct antecedent basis. See also claim 15 for a similar correction.

c. Regarding claim 6, the claim language as it currently reads that the "segment cache" comprises itself by being located within the "instruction segment system." Please correct this claim language to more clearly define the relationship between the "segment cache" and the "instruction segment system."

d. Regarding claim 9, the limitation "first and second instruction segments" lacks antecedent basis. Please add the limitation "first" before defining "an instruction segment" in claim 8 so as to provide the correct antecedent basis.

e. Regarding claim 16, the limitation "the instruction segment" is cited. However, the parent claims 14 and 15 only provide basis for "instruction segments." Please correct the claim language to provide correct antecedent basis for a singular "instruction segment." See also claims 17 and 18 for similar corrections.

f. Regarding claim 19, the limitation "the extended segment cache system" lacks antecedent basis. Please correct the claim's parent claims to define this claim language.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

6. Claims 1-19 are rejected under 35 U.S.C. 102(a) as anticipated by Jourdan et al., *Extended Block Cache*.

7. The applied reference has common inventors, Jourdan and Ronen, with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(a). This rejection under 35 U.S.C. 102(a) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

8. Regarding claim 1, Jourdan has taught an instruction segment comprising a plurality of instructions stored in sequential positions of a cache line (see Sec. 2.3 lines 27-28) in reverse order (see Sec. 3.4).

9. Regarding claim 2, Jourdan has taught the instruction segment of claim 1, wherein the instruction segment is an extended block (see Secs. 3 and 3.1).

10. Regarding claim 3, Jourdan has taught the instruction segment of claim 1, wherein the instruction segment is a trace (see Sec. 2.3 lines 2-5).

11. Regarding claim 4, Jourdan has taught the instruction segment of claim 1, wherein the instruction segment is a basic block (see Sec 3.1 lines 26-39).

12. Regarding claim 5, Jourdan has taught a segment cache for a front-end system in a processor, comprising a plurality of cache entries to store instruction segments (see Sec. 3) in reverse program order (see Sec. 3.4).

13. Regarding claim 6, Jourdan has taught the segment cache of claim 5, further comprising:

a. An instruction storage system (see "cache" of Fig.6);

b. An instruction segment system, comprising:

I. A fill unit (see Fig.4, 6) provided in communication with the instruction cache system (see Sec.3.5 lines 1-2);

II. Wherein the segment cache is included within the instruction segment system (see Fig.6);

III. And a selector coupled to the output of the instruction cache system and to an output of the segment cache (see "multiplexer" of Fig.6).

14. Regarding claim 7, Jourdan has taught the front-end system of claim 6, wherein the instruction segment system further comprises a segment predictor provided in communication with the segment cache (see Fig.4 and Sec.3.5 lines 27-30).

15. Regarding claim 8, Jourdan has taught a method for storing instruction segments in a processor (see Sec.3), comprising:

a. Building an instruction segment based on program flow (see Sec.3.4);

b. Storing the instruction segment in a cache in reverse program order (see Sec.3.4).

16. Regarding claim 9, Jourdan has taught the method of claim 8, further comprising:

a. Building a second instruction segment based on program flow (see Sec.3.3);

b. If the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment (see Sec.3.3).

17. Regarding claim 10, Jourdan has taught the method of claim 9, wherein the extending comprises storing the non-overlapping instructions in the cache in reverse program order in

successive cache positions adjacent to the instructions from the first instruction segment (see Sec.3.4).

18. Regarding claim 11, Jourdan has taught the instruction segment of claim 8, wherein the instruction segment is an extended block (see Secs. 3 and 3.1).

19. Regarding claim 12, Jourdan has taught the instruction segment of claim 8, wherein the instruction segment is a trace (see Sec. 2.3 lines 2-5).

20. Regarding claim 13, Jourdan has taught the instruction segment of claim 8, wherein the instruction segment is a basic block (see Sec 3.1 lines 26-39).

21. Regarding claim 14, Jourdan has taught a processing engine, comprising:

a. A front-end stage to build and store instruction segments (see Sec.3) in reverse program order (see Sec.3.4);

b. An execution unit in communication with the front end stage (see "execution block" of Sec.1).

22. Regarding claim 15, Jourdan has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. An instruction storage system (see "cache" of Fig.6);

b. An instruction segment system, comprising:

I. A fill unit (see Fig.4, 6) provided in communication with the instruction cache system (see Sec.3.5 lines 1-2);

II. A segment cache (see Fig.6);

III. A selector coupled to the output of the instruction cache system and to an output of the segment cache (see "multiplexer" of Fig.6).

23. Regarding claim 16, Jourdan has taught the instruction segment of claim 15, wherein the instruction segment is an extended block (see Secs. 3 and 3.1).

24. Regarding claim 17, Jourdan has taught the instruction segment of claim 15, wherein the instruction segment is a trace (see Sec. 2.3 lines 2-5).

25. Regarding claim 18, Jourdan has taught the instruction segment of claim 15, wherein the instruction segment is a basic block (see Sec 3.1 lines 26-39).

26. Regarding claim 19, Jourdan has taught the processing engine of claim 15, wherein the extended segment cache system further comprises a segment predictor provided in communication with the segment cache (see Fig.4 and Sec.3.5 lines 27-30).

Claim Rejections - 35 USC § 103

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

28. Claims 1-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peleg et al., U.S. Patent No. 5,381,533.

29. Regarding claim 1, Peleg has taught an instruction segment comprising a plurality of instructions stored in sequential positions of a cache line. Peleg has not taught the instructions being stored in reverse order.

30. However, the differences between the claim and Peleg are only found in the non-functional data stored on the article of manufacture, namely the cache. Data identifying the program order is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store instruction segments in reverse program order because the data of the instruction segments does not functionally relate to the substrate of the article of manufacture and merely labeling the data differently from that in the prior art would have been obvious. See Gulack cited above.

31. Regarding claims 2-4, Peleg has taught the instruction segment as shown in claim 1 above, but has not specifically taught the instruction segment being an extended block, a trace, or a basic block.

32. However, the differences between the claim and Peleg are only found in the non-functional data stored on the article of manufacture, namely the cache. Identifying the data set as a specific type is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to identify the instruction segment being stored as either an extended block, a trace, or a basic block because the data of the instruction segments does not functionally related to the substrate of the article of manufacture

and merely labeling the data set differently from that in the prior art would have been obvious. See Gulack cited above.

33. Claims 5-8 and 11-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing.*

34. Regarding claim 5, Patel has taught a segment cache (see "trace cache" of Fig.1) for a front-end system in a processor, comprising a plurality of cache entries to store instruction segments (see Sec.1 lines 10-18). Patel has not taught the instructions being stored in reverse order.

35. However, the differences between the claim and Patel are only found in the non-functional data stored on the article of manufacture, namely the cache. Data identifying the program order is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store instruction segments in reverse program order because the data of the instruction segments does not functionally relate to the substrate of the article of manufacture and merely labeling the data differently from that in the prior art would have been obvious. See Gulack cited above.

36. Regarding claim 6, Patel has taught the segment cache of claim 5, further comprising:

a. An instruction storage system (see "instruction cache" of Fig.1);

b. An instruction segment system, comprising:

I. A fill unit (see Fig.1) provided in communication with the instruction cache system (see Fig.1);

II. Wherein the segment cache (see "trace cache" of Fig.1) is included within the instruction segment system (see Fig.1);

III. And a selector coupled to the output of the instruction cache system and to an output of the segment cache (see "selection logic" of Fig.1).

37. Regarding claim 7, Patel has taught the front-end system of claim 6, wherein the instruction segment system further comprises a segment predictor (see "multiple branch predictor" of Fig.1) provided in communication with the segment cache. When the multiple-branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Sec.3 paragraphs 3-5).

38. Regarding claim 8, Patel has taught a method for storing instruction segments in a processor, comprising:

a. Building an instruction segment based on program flow (see Sec.1 lines 10-18);

b. Storing the instruction segment in a cache (see Sec.1 lines 10-18);

39. Patel has not taught the storing of the instruction segment in reverse order.

40. However, the differences between the claim and Patel are only found in the non-functional data stored on the article of manufacture, namely the cache. Data identifying the program order is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In*

re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store instruction segments in reverse program order because the data of the instruction segments does not functionally relate to the substrate of the article of manufacture and merely labeling the data differently from that in the prior art would have been obvious. See Gulack cited above.

41. Regarding claims 11-13, Patel has taught the instruction segment as shown in claim 8 above, but has not specifically taught the instruction segment being an extended block, a trace, or a basic block.

42. However, the differences between the claim and Patel are only found in the non-functional data stored on the article of manufacture, namely the cache. Identifying the data set as a specific type is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to identify the instruction segment being stored as either an extended block, a trace, or a basic block because the data of the instruction segments does not functionally related to the substrate of the article of manufacture and merely labeling the data set differently from that in the prior art would have been obvious. See Gulack cited above.

43. Regarding claim 14, Patel has taught a processing engine, comprising:

a. A front-end stage to build and store instruction segments (see Sec.1 lines 10-18);

b. An execution unit in communication with the front end stage (see "HPS Execution Core" of Fig. 1 and Sec.3 paragraph 6).

44. Patel has not taught the instructions being built and stored in reverse program order.

45. However, the differences between the claim and Patel are only found in the non-functional data stored on the article of manufacture, namely the cache. Data identifying the program order is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store instruction segments in reverse program order because the data of the instruction segments does not functionally relate to the substrate of the article of manufacture and merely labeling the data differently from that in the prior art would have been obvious. See Gulack cited above.

46. Regarding claim 15, Patel has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. An instruction storage system (see "instruction cache" of Fig. 1);

b. An instruction segment system, comprising:

I. A fill unit (see Fig. 1) provided in communication with the instruction cache system (see Fig. 1);

II. A segment cache (see "trace cache" of Fig. 1);

III. A selector (see "selection logic" of Fig. 1) coupled to the output of the instruction cache system (see "instruction cache" of Fig. 1) and to an output of the segment cache (see "trace cache" of Fig. 1).

47. Regarding claims 16-18, Patel has taught the instruction segment as shown in claim 15 above, but has not specifically taught the instruction segment being an extended block, a trace, or a basic block.

48. However, the differences between the claim and Patel are only found in the non-functional data stored on the article of manufacture, namely the cache. Identifying the data set as a specific type is not functionally related to the substrate of the article of manufacture. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, *see Cf. In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to identify the instruction segment being stored as either an extended block, a trace, or a basic block because the data of the instruction segments does not functionally related to the substrate of the article of manufacture and merely labeling the data set differently from that in the prior art would have been obvious. See Gulack cited above.

49. Regarding claim 19, Patel has taught the front-end system of claim 15, wherein the extended segment cache system further comprises a segment predictor (see "multiple branch predictor" of Fig. 1) provided in communication with the segment cache. When the multiple-branch predictor is coupled with the trace cache and mediated by the selection logic, it

effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Sec.3 paragraphs 3-5).

50. Claims 1-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peleg et al., U.S. Patent No. 5,381,533, and further in view of Peled et al., U.S. Patent 6,076,144.

51. Regarding claim 1, Peleg has taught an instruction segment comprising a plurality of instructions stored in sequential positions of a cache line. Peleg has not taught the instructions being stored in reverse order.

52. However, Peled has taught that it is desirable to allow extension of an instruction trace to have multiple entry points (see Col.1 lines 60-63) to reduce cache redundancy. One of ordinary skill in the art would have recognized that updating an instruction segment to contain multiple entry points requires the front (head) of the segment to be extended, as the back (tail) of the segment is always a single conditional branch instruction. In order for the extension to take place, either the instructions need to be moved to higher addresses so that the pointer to the front of the segment is not changed, or the pointer to the front of the segment needs to be updated to reflect the new front, or both, in order to keep the instructions within the segment in sequential order. One of ordinary skill in the art would then have recognized that the head and tail pointers are arbitrary, reversing the addressing scheme by switching the head and tail pointers so that tail stays at a stationary low address and the head grows into the high addresses, effectively reversing the order the instructions of the segment are stored, will allow the pointer to the segment to remain unchanged, and wont require the instructions to be moved when the segment is extended. It is also well known in the art that a goal of cache design is to provide access to the widest variety of frequently used instructions or instruction sequences in as little time with as little work

as possible. Therefore it would have been obvious to one of ordinary skill in the art to store the instructions of the instruction segment in reverse order so that cache redundancy can be reduced with as little processing effort as possible.

53. Regarding claim 2, Peleg has taught the instruction segment of claim 1, but has not taught the instruction segment being an extended block. Peled has taught trace segments which have multiple entry points with a single exit which allow redundant code segments to be eliminated (see Col.1 lines 60-63, Col. 4 lines 13-37 and Fig.3). One of ordinary skill in the art would have recognized that is desirable to reduce redundancy within in a trace cache so that the most different traces can be stored for fast access. Therefore, one of ordinary skill in the art would have found it obvious to store instruction segments in such a manner as to allow multiple entries into the traces, allowing redundant sections of trace to be eliminated.

54. Regarding claim 3, Peleg has taught the instruction segment of claim 1, wherein the instruction segment is a trace (see Col.4 lines 6-10).

55. Regarding claim 4, Peleg has taught the instruction segment of claim 1, wherein the instruction segment is a basic block (see Col.2 lines 1-5 and Col.4 lines 6-15).

56. Claims 5-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing*, and further in view of Peled et al., U.S. Patent No. 6,076,144

57. Regarding claim 5, Patel has taught a segment cache (see "trace cache" of Fig.1) for a front-end system in a processor, comprising a plurality of cache entries to store instruction segments (see Sec.1 lines 10-18). Patel has not taught the instructions being stored in reverse order.

58. However, Peled has taught that it is desirable to allow extension of an instruction trace to have multiple entry points (see Col.1 lines 60-63) to reduce cache redundancy. One of ordinary skill in the art would have recognized that updating an instruction segment to contain multiple entry points requires the front (head) of the segment to be extended, as the back (tail) of the segment is always a single conditional branch instruction. In order for the extension to take place, either the instructions need to be moved to higher addresses so that the pointer to the front of the segment is not changed, or the pointer to the front of the segment needs to be updated to reflect the new front, or both, in order to keep the instructions within the segment in sequential order. One of ordinary skill in the art would then have recognized that the head and tail pointers are arbitrary, reversing the addressing scheme by switching the head and tail pointers so that tail stays at a stationary low address and the head grows into the high addresses, effectively reversing the order the instructions of the segment are stored, will allow the pointer to the segment to remain unchanged, and wont require the instructions to be moved when the segment is extended. It is also well known in the art that a goal of cache design is to provide access to the widest variety of frequently used instructions or instruction sequences in as little time with as little work as possible. Therefore it would have been obvious to one of ordinary skill in the art to store the instructions of the instruction segment in reverse order so that cache redundancy can be reduced with as little processing effort as possible.

59. Regarding claim 6, Patel has taught the segment cache of claim 5, further comprising:

a. An instruction storage system (see "instruction cache" of Fig.1);

b. An instruction segment system, comprising:

I. A fill unit (see Fig.1) provided in communication with the instruction cache system (see Fig.1);

II. Wherein the segment cache (see "trace cache" of Fig.1) is included within the instruction segment system (see Fig.1);

III. And a selector coupled to the output of the instruction cache system and to an output of the segment cache (see "selection logic" of Fig.1).

60. Regarding claim 7, Patel has taught the front-end system of claim 6, wherein the instruction segment system further comprises a segment predictor (see "multiple branch predictor" of Fig.1) provided in communication with the segment cache. When the multiple-branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Sec.3 paragraphs 3-5).

61. Regarding claim 8, Patel has taught a method for storing instruction segments in a processor, comprising:

a. Building an instruction segment based on program flow (see Sec.1 lines 10-18);

b. Storing the instruction segment in a cache (see Sec.1 lines 10-18);

62. Patel has not taught the storing of the instruction segment in reverse order.

63. However, Peled has taught that it is desirable to allow extension of an instruction trace to have multiple entry points (see Col.1 lines 60-63) to reduce cache redundancy. One of ordinary skill in the art would have recognized that updating an instruction segment to contain multiple entry points requires the front (head) of the segment to be extended, as the back (tail) of the segment is always a single conditional branch instruction. In order for the extension to take

place, either the instructions need to be moved to higher addresses so that the pointer to the front of the segment is not changed, or the pointer to the front of the segment needs to be updated to reflect the new front, or both, in order to keep the instructions within the segment in sequential order. One of ordinary skill in the art would then have recognized that the head and tail pointers are arbitrary, reversing the addressing scheme by switching the head and tail pointers so that tail stays at a stationary low address and the head grows into the high addresses, effectively reversing the order the instructions of the segment are stored, will allow the pointer to the segment to remain unchanged, and wont require the instructions to be moved when the segment is extended. It is also well known in the art that a goal of cache design is to provide access to the widest variety of frequently used instructions or instruction sequences in as little time with as little work as possible. Therefore it would have been obvious to one of ordinary skill in the art to store the instructions of the instruction segment in reverse order so that cache redundancy can be reduced with as little processing effort as possible.

64. Regarding claim 9, Patel has taught the method of claim 8, but has not taught the building and combining of a second instruction segment if it overlaps with a first instruction segment.

65. Peled has taught building a second instruction segment based on program flow and if the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment (see Col.4 lines 13-37 and Fig.3), reducing the degree of code redundancy in the trace cache (see Col.1 lines 60-63). One of ordinary skill in the art would have recognized that is desirable to reduce redundancy within in a trace cache so that the most different traces can be stored for fast access. Therefore, one of ordinary skill in the art would have found it obvious to extend existing instruction segments to

include non-overlapping instructions from a second instruction segment in order to reduce instruction redundancy within a trace cache.

66. Regarding claim 10, Patel in view of Peled has taught the method of claim 9 including the storing of instructions in reverse program order, but has not taught where the extending comprises storing the non-overlapping instructions in the cache in successive cache positions adjacent to the instructions from the first instruction segment.

67. Peled has taught building a second instruction segment based on program flow and if the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment (see Col.4 lines 13-37 and Fig.3), reducing the degree of code redundancy in the trace cache (see Col.1 lines 60-63). One of ordinary skill in the art would have recognized that is desirable to reduce redundancy within in a trace cache so that the most different traces can be stored for fast access. Therefore, one of ordinary skill in the art would have found it obvious to extend existing instruction segments to include non-overlapping instructions from a second instruction segment in the cache in successive cache positions adjacent to instructions from the first instruction segment in order to reduce instruction redundancy within a trace cache.

68. Regarding claim 11, Patel has taught the method of claim 8, wherein the instruction segment is a basic block or a trace, but has not taught the instruction segment being an extended block. Peled has taught trace segments which have multiple entry points with a single exit which allow redundant code segments to be eliminated (see Col.1 lines 60-63, Col. 4 lines 13-37 and Fig.3). One of ordinary skill in the art would have recognized that is desirable to reduce redundancy within in a trace cache so that the most different traces can be stored for fast access.

Therefore, one of ordinary skill in the art would have found it obvious to store instruction segments in such a manner as to allow multiple entries into the traces, allowing redundant sections of trace to be eliminated.

69. Regarding claim 12, Patel has taught the method of claim 8, wherein the instruction segment is a trace (see Sec.1 paragraph 4).

70. Regarding claim 13, Patel has taught the method of claim 8, wherein the instruction segment is a basic block (see Sec.1 lines 10-18).

71. Regarding claim 14, Patel has taught a processing engine, comprising:

a. A front-end stage to build and store instruction segments (see Sec.1 lines 10-18);

b. An execution unit in communication with the front end stage (see "HPS Execution Core" of Fig.1 and Sec.3 paragraph 6).

72. Patel has not taught the instructions being built and stored in reverse program order.

73. However, Peled has taught that it is desirable to allow extension of an instruction trace to have multiple entry points (see Col.1 lines 60-63) to reduce cache redundancy. One of ordinary skill in the art would have recognized that updating an instruction segment to contain multiple entry points requires the front (head) of the segment to be extended, as the back (tail) of the segment is always a single conditional branch instruction. In order for the extension to take place, either the instructions need to be moved to higher addresses so that the pointer to the front of the segment is not changed, or the pointer to the front of the segment needs to be updated to reflect the new front, or both, in order to keep the instructions within the segment in sequential order. One of ordinary skill in the art would then have recognized that the head and tail pointers are arbitrary, reversing the addressing scheme by switching the head and tail pointers so that tail

stays at a stationary low address and the head grows into the high addresses, effectively reversing the order the instructions of the segment are stored, will allow the pointer to the segment to remain unchanged, and wont require the instructions to be moved when the segment is extended. It is also well known in the art that a goal of cache design is to provide access to the widest variety of frequently used instructions or instruction sequences in as little time with as little work as possible. Therefore it would have been obvious to one of ordinary skill in the art to store the instructions of the instruction segment in reverse order so that cache redundancy can be reduced with as little processing effort as possible.

74. Regarding claim 15, Patel has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. An instruction storage system (see "instruction cache" of Fig. 1);

b. An instruction segment system, comprising:

I. A fill unit (see Fig. 1) provided in communication with the instruction cache system (see Fig. 1);

II. A segment cache (see "trace cache" of Fig. 1);

III. A selector (see "selection logic" of Fig. 1) coupled to the output of the instruction cache system (see "instruction cache" of Fig. 1) and to an output of the segment cache (see "trace cache" of Fig. 1).

75. Regarding claim 16, Patel has taught the method of claim 15, wherein the instruction segment is a basic block or a trace, but has not taught the instruction segment being an extended block. Peled has taught trace segments which have multiple entry points with a single exit which allow redundant code segments to be eliminated (see Col. 1 lines 60-63, Col. 4 lines 13-37 and

Fig.3). One of ordinary skill in the art would have recognized that is desirable to reduce redundancy within in a trace cache so that the most different traces can be stored for fast access. Therefore, one of ordinary skill in the art would have found it obvious to store instruction segments in such a manner as to allow multiple entries into the traces, allowing redundant sections of trace to be eliminated.

76. Regarding claim 17, Patel has taught the method of claim 15, wherein the instruction segment is a trace (see Sec.1 paragraph 4).

77. Regarding claim 18, Patel has taught the method of claim 15, wherein the instruction segment is a basic block (see Sec.1 lines 10-18).

78. Regarding claim 19, Patel has taught the front-end system of claim 15, wherein the extended segment cache system further comprises a segment predictor (see "multiple branch predictor" of Fig.1) provided in communication with the segment cache. When the multiple-branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Sec.3 paragraphs 3-5).

Conclusion

79. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

80. Peled et al, U.S. Patent No. 6,073,213, has taught a method for creating instruction segments that allow multiple entry points into them.

81. Bala et al, U.S. Patent No. 6,351,844, has taught the reordering of code traces in a trace cache to reduce redundancy.

82. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (703) 305-5864. The examiner can normally be reached on Mon.-Fri. 7:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
9/22/2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100